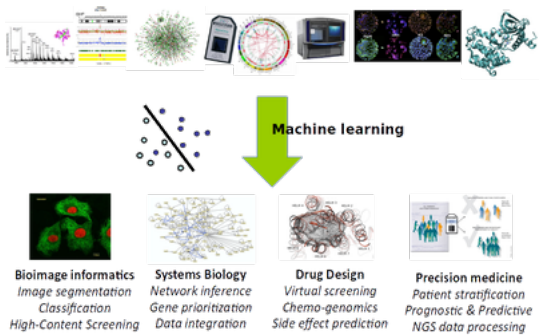
Outline

Are machines learning ?

- Convergence of math/info/computer science research
- Inspired by research in neuroscience and cognition (and science fiction)
- Contemporary to high throughput data acquisition
- Two basic ingredients: data and algorithms

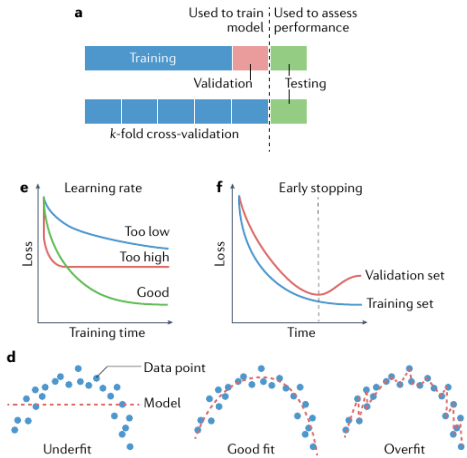
Machine learning in Biology

- Data have grown in complexity and size in all fields of biology
- Data are multimodal: sequences, structures, spectra, images, molecular, clinical, evolutionary
- Impossible to handle the analysis by descriptive methods only
- Reproducible research
- Machine Learning in Biology has become a field on its own



The quantitative shift [?]

- knowledge transfer of basic concepts in ML for biologists
- training / testing
- over-fitting / under-fitting
- Linear / non linear
- Interpretability of ML methods
- Computational complexity / time



The two sides of Machine Learning

Supervised Learning

- Observe $(y_1, x_1), \ldots, (y_n, x_n)$
- Construct a predictor $f : \mathcal{X} \to \mathcal{Y}$
- Define a loss function $\ell(y, f(x))$ to score predictions
- Minimize the generalization error (new y ?)

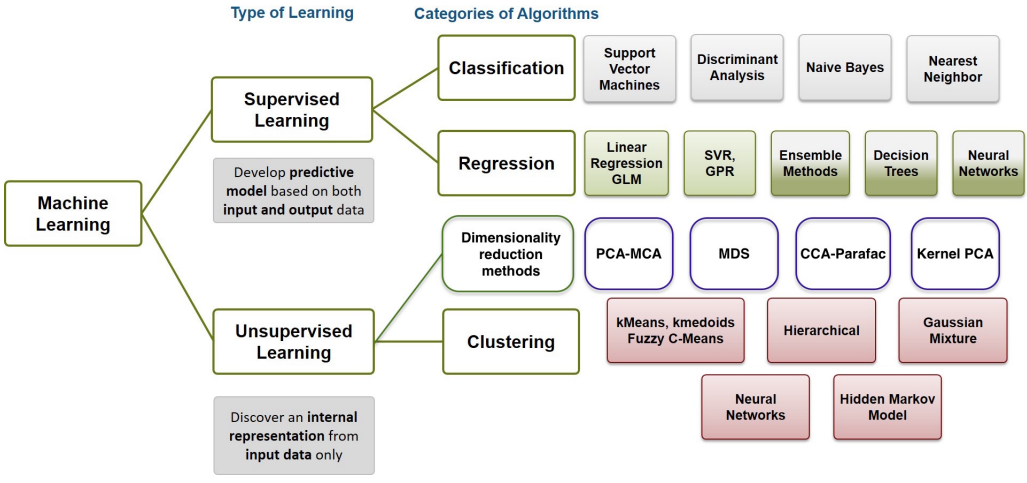
$\to$ Regression, classification

Non-Supervised Learning

- Observe $(x_1, \ldots, x_n)$
- Describe the structure of X without external information
- Group individuals ? Group variables ?
- Loss is more difficult to define

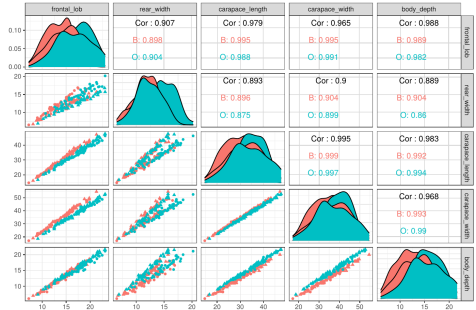
$\to$ Dimension Reduction, Clustering

Rough typology of ML methods



The purpose of Dimension Reduction

- Visualization (> 2 variables)
- Multivariate analysis (beyond pairwise)
- Summary of the data
- Redundancy
- Reduce the data for downstream methods



Crabs dataset ($n = 200, p = 8$)

An unprecedented challenge

- Genomics was precursor for data representation and visualization
- Gene Expression data $\sim 30,000$ variables
- Recent single-cell technologies: up to 10^6 cells

Publication	cells	tissue	Seq. protocol
Cadwell et al. (2016)	46	visual cortex	Smart-seq2
Tasic et al. (2016)	1,679	visual cortex	SMARTer
Macosko et al. (2015)	44,808	retina	Drop-seq
10x Genomics	1,306,127	brain cells	10x Gen.Chrom.

- Dimension reduction is mandatory for any analysis (clustering, visualization, inference)

High-dimensional data

$$X_i^j = \text{measurement for variable } j \text{ on individual } i$$

$$\mathbf{X}_{n\times p} = \begin{bmatrix} & & & & & & \\ & & & X_i^j & & & \\ & & & & & & \end{bmatrix} \begin{matrix} 1 \\ \vdots \\ n \end{matrix} \Big\} \text{invidivuals}$$

$$\underbrace{1 \quad \dots \quad \dots \quad \dots \quad \dots \quad p}_{\text{variables}}$$

- **Ideal case:** n grows and $\gg p$
- **High dimension:** p grows and $\gg n$
- **Big Data:** n and p grow

Outline

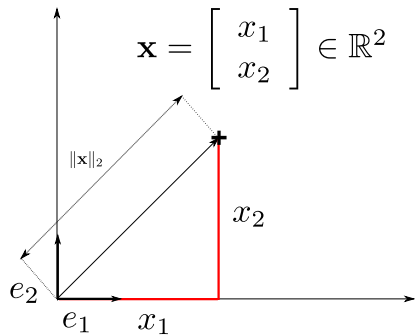
Vectors of $\mathbb{R}^d$

- $\mathbf{x}$ is a vector of $\mathbb{R}^d$ defined by a $n-$uplet $(x_1, \ldots, x_d)$ (coordinates)
- Considering the canonical basis ($d = 2$):

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Its coordinates corresponds to a decomposition on the unitary basis:

$$\mathbf{x} \in \mathbb{R}^2, \quad \mathbf{x} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

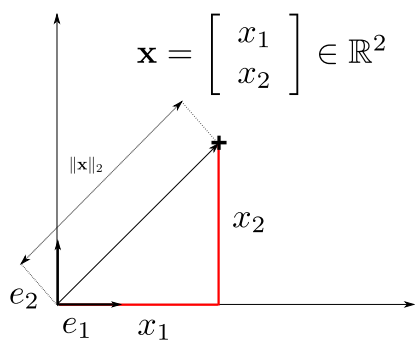


Vectors of $\mathbb{R}^d$

- A vector is a point in a space (here $\mathbb{R}^2$)
- Generalize for vectors of $\mathbb{R}^d$

$$\mathbf{x} \in \mathbb{R}^d, \quad \mathbf{x} = \sum_{h=1}^{d} x_h \mathbf{e}_h$$

- By default, $\mathbf{x}$ is a column vector, $\mathbf{x}'$ its transpose
- Concatenate p column vectors $[\mathbf{x}_1, \ldots, \mathbf{x}_p]$.

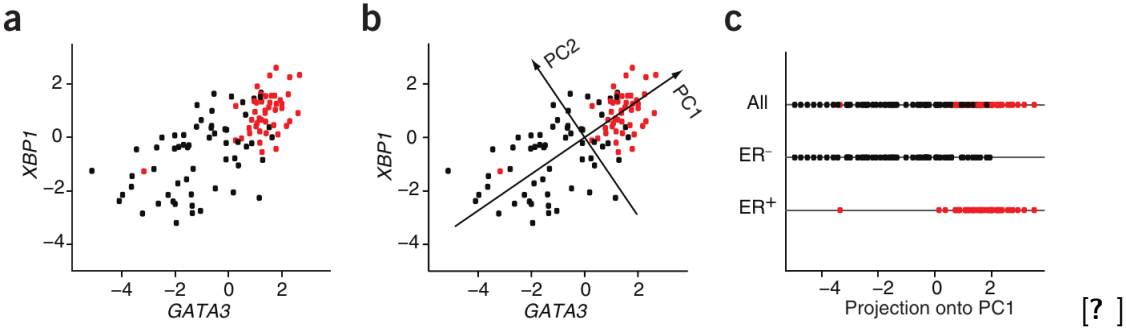


Expression of 105 breast tumor samples ER(+/-)

Measure of the expression of two genes XBP1 and GATA3 for $n = 105$ samples

Each measure is denoted by $\mathbf{x}_i = \left[x_i^{\text{GATA3}}, x_i^{\text{XBP1}} \right]$

Each point is a vector of $\mathbb{R}^2$

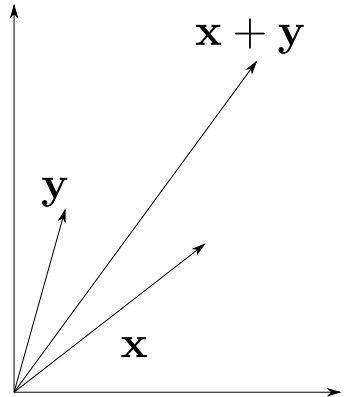


[?]

Vectors of $\mathbb{R}^d$ and basic operations: Addition

$$\begin{aligned} \mathbf{x}+\mathbf{y} &= \begin{bmatrix} x_1+y_1 \\ \vdots \\ x_d+y_d \end{bmatrix} \\ (\mathbf{x}+\mathbf{y}) &= (\mathbf{y}+\mathbf{x}), \\ (\mathbf{x}+\mathbf{y})+\mathbf{z} &= \mathbf{x}+(\mathbf{y}+\mathbf{z}) \end{aligned}$$

Associative, Commutative



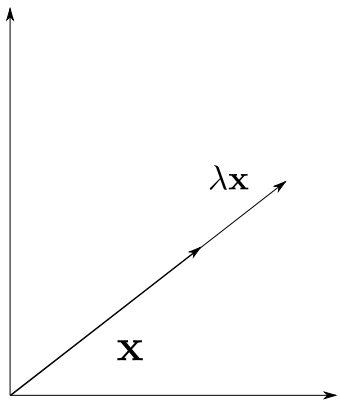
Vectors of $\mathbb{R}^d$ and basic operations: Multiplication by a scalar

$$\forall \lambda \in \mathbb{R},\ \lambda \mathbf{x} = \begin{bmatrix} \lambda x_1 \\ \vdots \\ \lambda x_d \end{bmatrix}$$

$$\lambda(\mathbf{x} + \mathbf{y}) = \lambda \mathbf{x} + \lambda \mathbf{y},$$

Linear Combination

$$(\lambda_1 + \lambda_2)\mathbf{x} = \lambda_1 \mathbf{x} + \lambda_2 \mathbf{x}$$



Dot Product between vectors

- The dot product $\bullet$ between two vectors is defined by the sum of the products of all components

$$\mathbf{x} \bullet \mathbf{y} = \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}'\mathbf{y} = \sum_{i=1}^{d} x_i y_i, \quad \langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle, \quad \langle \mathbf{x}, \mathbf{y} + \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{z} \rangle$$

- The dot product between two vectors is a scalar
- Basic properties:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle, \quad \langle \mathbf{x}, \mathbf{y} + \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{z} \rangle, \quad \lambda \langle \mathbf{x}, \mathbf{y} \rangle = \langle \lambda \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, \lambda \mathbf{y} \rangle$$

Norm of a vector and basic properties

- The euclidean norm (length of a vector)

$$\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}'\mathbf{x} = \sum_{i=1}^{d} x_i^2$$

- Non negativity : $\|\mathbf{x}\|_2 \geq 0$
- Definiteness : $\|\mathbf{x}\|_2 = 0 \leftrightarrow \mathbf{x} = 0$
- Triangle Inequality : $\|\mathbf{x} + \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2$
- Homogeneity : $\|\lambda \times \mathbf{x}\|_2 = |\lambda| \times \|\mathbf{x}\|_2, \quad \lambda \in \mathbb{R}$

Principal norms used in Machine Learning

- L^1 norm or Manhattan norm:

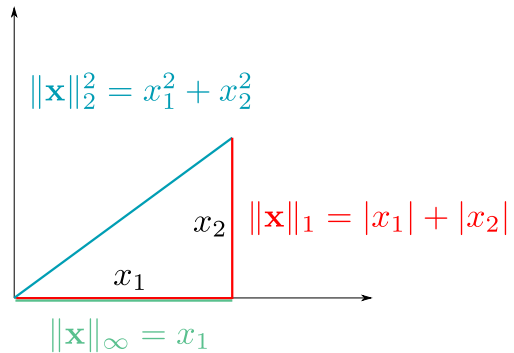
$$\|\mathbf{x}\|_1 = \sum_{i=1}^{d} |x_i|$$

- L^2 norm or Euclidian norm:

$$\|\mathbf{x}\|_2^2 = \sum_{i=1}^{d} x_i^2$$

- L^∞ norm or sup-norm:

$$\|\mathbf{x}\|_\infty = \max_{i=1,\ldots,d} \left(|x_i|\right)$$



There are different ways to measure the norm of a vector

From norms to distances between vectors

- L^1 distance or Manhattan distance:

$$d_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1 = \sum_{i=1}^{n} |x_i - y_i|$$

- L^2 distance or Euclidean distance:

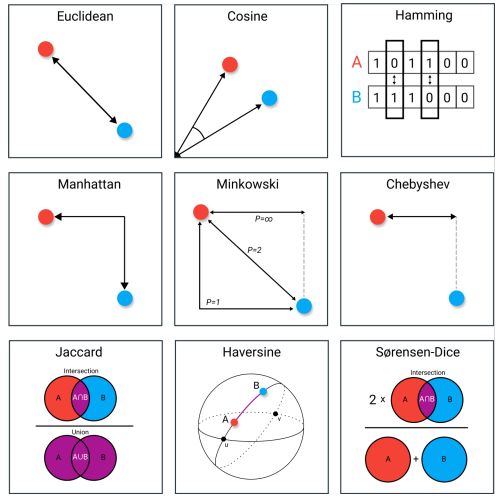
$$d_2^2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2 = \sum_{i=1}^{n} (x_i - y_i)^2$$

- L^∞ distance or sup-distance:

$$d_\infty(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_\infty = \max_{i=1,\dots,n} \Big(|x_i - y_i| \Big)$$

What drives the choice of a distance ?

- L^1 distance or Manhattan distance:
  - Adapted to discrete inputs
  - Robust to outliers
  - Non differentiable
- L^2 distance or Euclidean distance:
  - Most common, differentiable
  - Sensitive to dimension and outliers
  - Sensitive to the scale of the different inputs
- L^∞ distance or sup-distance:
  - Applied in logistical problems
  - More specific, less used



https://towardsdatascience.com/

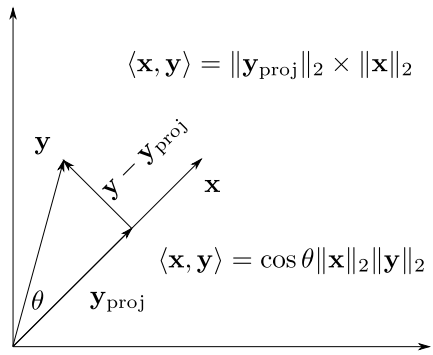
Dot product and orthogonal projection (1)

- The orthogonal projection of $\mathbf{y}$ on $\mathbf{x}$

$$\mathbf{y}_{\text{proj}} = \lambda \mathbf{x} \quad , \quad \text{colinearity}$$
$$\mathbf{y} - \mathbf{y}_{\text{proj}} \perp \mathbf{x} \quad , \quad \text{orthogonality of residuals}$$

- The proportionality coefficient is given by

$$\lambda = \frac{\langle \mathbf{y}, \mathbf{x} \rangle}{\|\mathbf{x}\|_2}$$



$$\|\mathbf{y}\|_2^2 = \|\mathbf{y}_{\text{proj}}\|_2^2 + \|\mathbf{y} - \mathbf{y}_{\text{proj}}\|_2^2$$

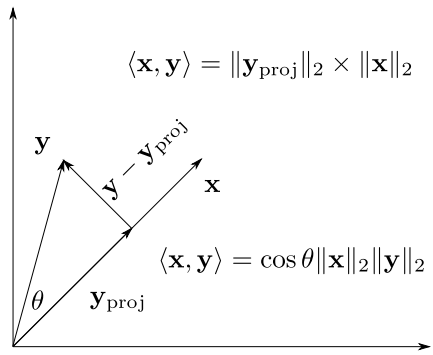
Dot product and orthogonal projection (2)

- Using trigonometry properties:

$$\cos\theta = \frac{\|\mathbf{y}_{proj}\|_2}{\|\mathbf{y}\|_2} = \lambda \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2}$$

- The dot product is the length of $\mathbf{x}$ times the length of the ortho. projection of $\mathbf{y}$
- Orthogonality :

$$\mathbf{x} \perp \mathbf{y} \leftrightarrow \langle \mathbf{y}, \mathbf{x} \rangle = 0$$



$$\|\mathbf{y}\|_2^2 = \|\mathbf{y}_{\text{proj}}\|_2^2 + \|\mathbf{y} - \mathbf{y}_{\text{proj}}\|_2^2$$

Outline

Vectors of individuals, vectors of variables

- $\mathbf{x}_i = (x_i^1, \ldots, x_i^p)' \in \mathbb{R}^p$ is the vector describing individual i with p different variables
- $\mathbf{x}^j = (x_i^j, \ldots, x_n^j)' \in \mathbb{R}^n$ is the vector of variable j on n different individuals

- The data is stored in a matrix $\mathbf{X}_{n \times p}$ such that

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1' \\ \vdots \\ \mathbf{x}_n' \end{bmatrix} = \begin{bmatrix} \mathbf{x}^1, \ldots, \mathbf{x}^p \end{bmatrix} = \underbrace{\left[\begin{array}{c|c|c|c|c|c|c} & & & & & & \\ \hline & & & x_i^j & & & \\ \hline & & & & & & \end{array}\right]}_{\text{variables} \in \mathbb{R}^n} \begin{matrix} 1 \\ \vdots \\ n \end{matrix} \Bigg\} \text{individuals} \in \mathbb{R}^p$$

$$\underbrace{1 \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad p}_{\text{variables} \in \mathbb{R}^n}$$

Centering a dataset (1)

- The empirical mean of $\mathbf{x}^j$:

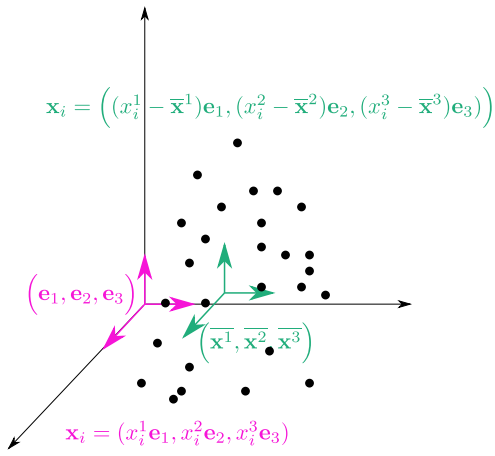
$$\overline{x}^j = \frac{1}{n}\sum_{i=1}^{n} x_i^j, \quad \overline{\mathbf{x}}_j = \overline{x}^j \times \mathbf{1}_n$$

- The empirical mean of $\mathbf{x}^j$ is its projection on the constant

$$\overline{x}^j = \frac{1}{n}(1,\ldots,1) \bullet \mathbf{x} = \frac{1}{n}\langle \mathbf{1}_n', \mathbf{x}^j \rangle$$

- The vector of means is the barycenter of the data

$$\overline{\mathbf{x}} = \left[\overline{x}^1, \ldots, \overline{x}^p\right]$$

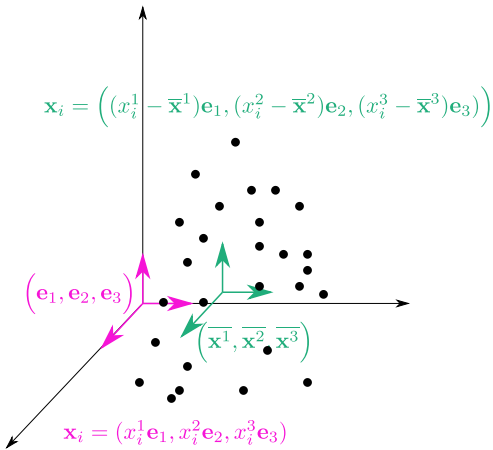


Centering a dataset: changing the origin

- Consists in removing the mean of each variable

$$\mathbf{X}_c = \left[\; \mathbf{x}^1 - \overline{\mathbf{x}}^1, \ldots, \mathbf{x}^p - \overline{\mathbf{x}}^p \; \right]$$

- Centering to avoid positional effects
- $\overline{\mathbf{x}}$ becomes the new origin



Scaling a dataset

- The empirical variance of $\mathbf{x}^j$:

$$\text{var}(\mathbf{x}^j) = \frac{1}{n}\sum_{i=1}^{n}(x_i^j - \overline{x}^j)^2$$

- It is the distance of variable $\mathbf{x}_j$ to its mean

$$\text{var}(\mathbf{x}^j) = \frac{1}{n}\|\mathbf{x}^j - \overline{\mathbf{x}}^j\|_2^2 = \frac{1}{n}\langle \mathbf{x}^j - \overline{\mathbf{x}}^j, \mathbf{x}^j - \overline{\mathbf{x}}^j \rangle = \frac{1}{n}\mathbf{x}_c^j \bullet \mathbf{x}_c^j$$

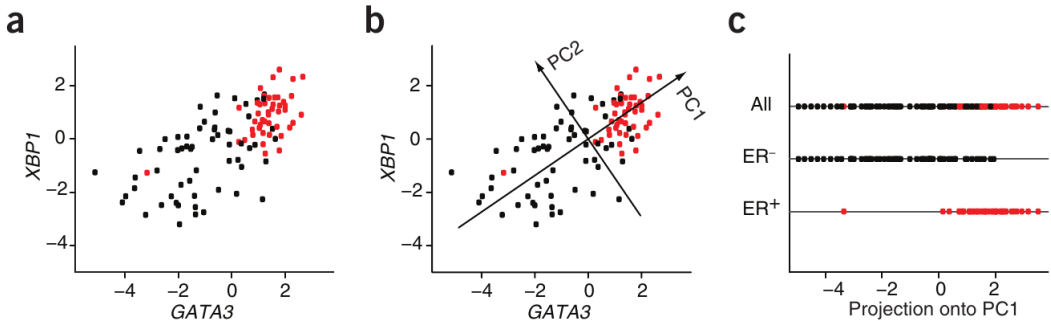
- The empirical variance is the length of the residuals (after centering)
- Scaling to standardize variables contributions (unitary variance)

$$\widetilde{\mathbf{X}}_c = \left[\frac{\mathbf{x}^1 - \overline{\mathbf{x}}^1}{\text{var}^{1/2}(\mathbf{x}^1)}, \ldots, \frac{\mathbf{x}^p - \overline{\mathbf{x}}^p}{\text{var}^{1/2}(\mathbf{x}^p)}\right]$$

Expression of 105 breast tumor samples ER(+/-)

The data matrix is

$$\mathbf{X}_c = \left[\mathbf{x}_c^{\text{GATA3}}, \mathbf{x}_c^{\text{XBP1}} \right]_{105 \times 2}$$



The expression of those 2 genes is very correlated: redundancy between columns

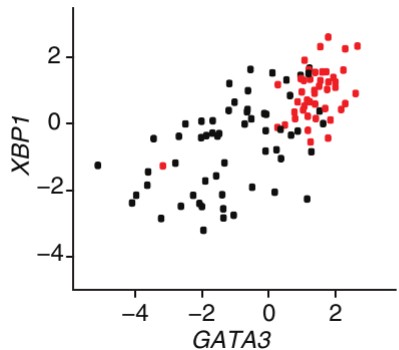
Covariance and Correlation between variables

- The empirical covariance between variables

$$\mathrm{c}(\mathbf{x}^j, \mathbf{x}^{j'}) = \frac{1}{n}\sum_{i=1}^{n}(x_i^j - \overline{\mathbf{x}}^j)(x_i^{j'} - \overline{\mathbf{x}}^{j'})$$

$$\mathrm{r}(\mathbf{x}^j, \mathbf{x}^{j'}) = \frac{\mathrm{c}(\mathbf{x}^j, \mathbf{x}^{j'})}{\sqrt{\mathrm{var}(\mathbf{x}^j)\,\mathrm{var}(\mathbf{x}^{j'})}}$$

- Quantifies the expected co-variations between variables
- If $\mathrm{r}(\mathbf{x}^j, \mathbf{x}^{j'}) \simeq 1$ the two variables provide the same information



Distance and covariance

- Between-variables distance:

$$\begin{aligned}\frac{1}{n}\|\mathbf{x}_c^j - \mathbf{x}_c^{j'}\|^2 &= \frac{1}{n}\|\mathbf{x}_c^j\|^2 + \frac{1}{n}\|\mathbf{x}_c^{j'}\|^2 - 2\frac{1}{n}\langle \mathbf{x}_c^j, \mathbf{x}_c^{j'}\rangle \\ &= \mathrm{var}(\mathbf{x}^j) + \mathrm{var}(\mathbf{x}^{j'}) - 2\,\mathrm{c}(\mathbf{x}^{j'}, \mathbf{x}^j)\end{aligned}$$

- Normalized distance using centered and scaled variables

$$\frac{1}{n}\|\widetilde{\mathbf{x}}_c^j - \widetilde{\mathbf{x}}_c^{j'}\|^2 = 2 - 2\,\mathrm{r}(\mathbf{x}^{j'}, \mathbf{x}^j)$$

- The correlation coefficient is a distance measure between variables:

$$\mathrm{r}(\mathbf{x}^{j'}, \mathbf{x}^j) = 1 - \frac{1}{2} \times \frac{1}{n}\|\widetilde{\mathbf{x}}_c^j - \widetilde{\mathbf{x}}_c^{j'}\|^2$$

Correlation and distance between variables

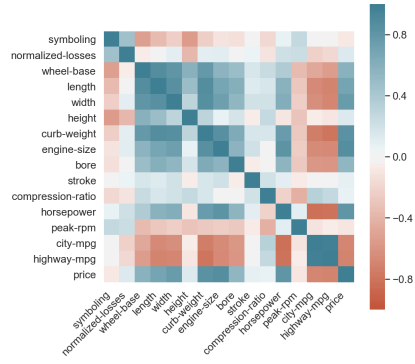
- Pairwise distance between variables

$$\mathbf{S} = \begin{bmatrix} \mathrm{c}(\mathbf{x}^1, \mathbf{x}^1) & \dots & \mathrm{c}(\mathbf{x}^{j'}, \mathbf{x}^j) \\ & \ddots & \\ \mathrm{c}(\mathbf{x}^j, \mathbf{x}^{j'}) & \dots & \mathrm{c}(\mathbf{x}^p, \mathbf{x}^p) \end{bmatrix}$$

- Normalized distance: correlation matrix

$$\mathbf{R} = \begin{bmatrix} \mathrm{r}(\mathbf{x}^1, \mathbf{x}^1) & \dots & \mathrm{r}(\mathbf{x}^{j'}, \mathbf{x}^j) \\ & \ddots & \\ \mathrm{r}(\mathbf{x}^j, \mathbf{x}^{j'}) & \dots & \mathrm{r}(\mathbf{x}^p, \mathbf{x}^p) \end{bmatrix}$$

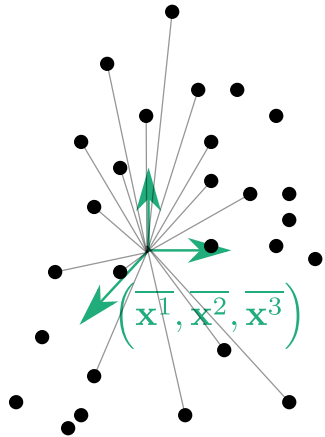
- Symmetric, invertible (semi definite positive)



Total Inertia of a dataset

The global variance of a dataset for centered variables

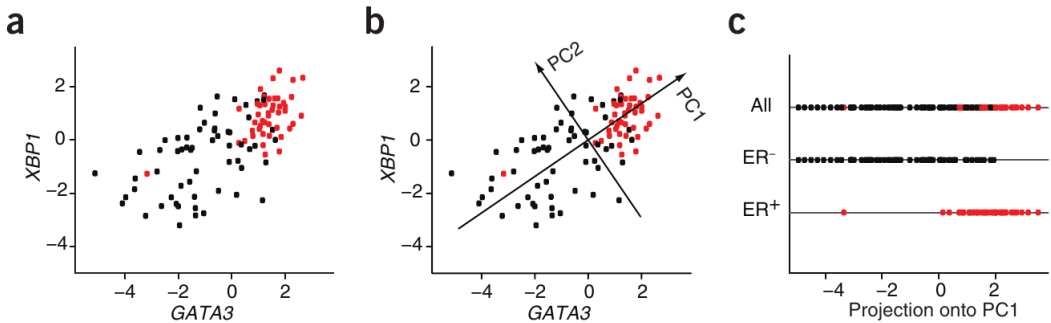
$$
\begin{aligned}
I_T(\mathbf{X}) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_i^j - \overline{x}^j)^2 \\
&= \sum_{j=1}^{p}\mathrm{var}(\mathbf{x}^j)
\end{aligned}
$$



Inertia of a dataset

To generalize the notion of dispersion to a complete dataset:

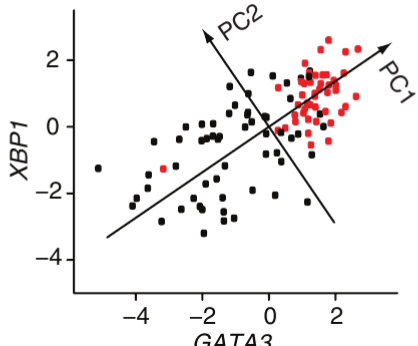
$$I_T(\mathbf{X}) \quad = \quad \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_i^j - \overline{x}^j)^2$$



Outline

Outline before PCA

- PCA is based on a change in coordinates
- Before performing PCA, focus on the rotation of a dataset
- Change coordinates from 2D to 2D, then generalize



From 2D to 2D with rotation

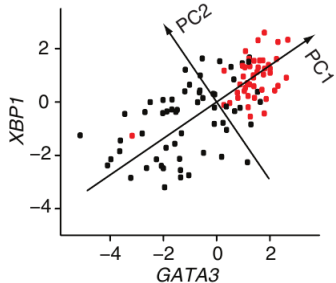
- Find new coordinates $\mathbf{Z}$ to better represent $\mathbf{X}$
- Define z_{1i} the new coordinates of individual i on axis 1 as linear combinations of the ancient coordinates

$$z_{1i} = v_{11}\widetilde{x}^1_{i,c} + v_{12}\widetilde{x}^2_{i,c}$$

- This operation resumes to a linear transform of $\mathbf{x}_i$ (old) to obtain $\mathbf{z}$ (new)

$$\mathbf{z}_{i1} = \widetilde{\mathbf{x}}_{i,c}\mathbf{v}_1$$

- How to determine $\mathbf{v}_1 = \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix}_{2\times 1}$?



New Coordinates

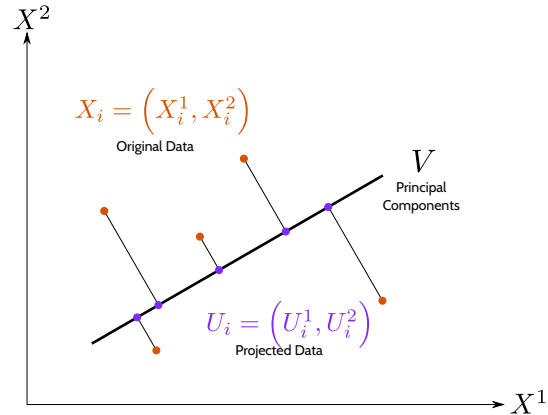
- In the example:

$$z_i^1 = 0.83 \times \text{GATA3}_i + 0.56 \times \text{XBP1}_i$$

- For the best representation of **X**

$$\widehat{v}_{11} = 0.83, \quad \widehat{v}_{12} = 0.56,$$

- Notation $\widehat{v}$ stands for optimized coordinates

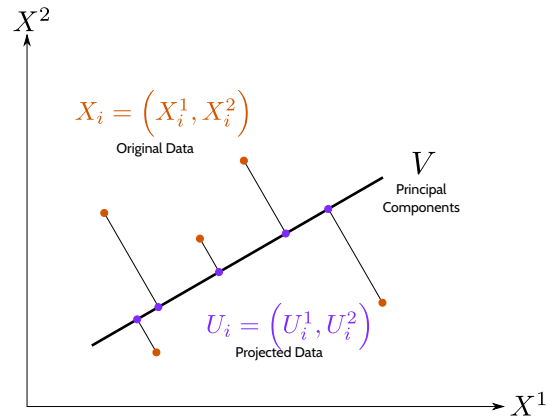


New coordinates in the matricial framework

- The coefficients are common to all individuals:

$$\begin{aligned} \mathbf{z}_1 &= v_{11}\widetilde{\mathbf{x}}_c^1 + v_{12}\widetilde{\mathbf{x}}_c^2 \\ &= \left[\begin{array}{cc} \widetilde{\mathbf{x}}_c^1 & \widetilde{\mathbf{x}}_c^2 \end{array}\right]_{n\times 2} \left[\begin{array}{c} v_{11} \\ v_{12} \end{array}\right]_{2\times 1} \\ \mathbf{z}_1 &= \widetilde{\mathbf{X}}_c \mathbf{v}_1 \end{aligned}$$

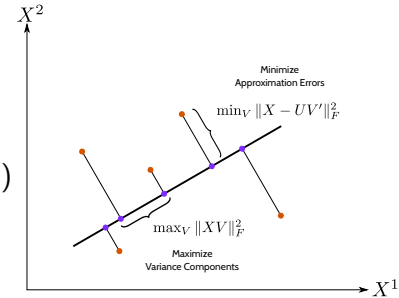
- Equation of a line with slope $\mathbf{v}_1$
- Centered data so no intercept



New coordinates in the matrix framework (1)

- First axis carries the biggest empirical variance

$$\begin{aligned}\text{var}(\mathbf{z}_1) &= \text{var}\left(\widetilde{\mathbf{X}}_c \mathbf{v}_1\right) \\ &= \text{var}\left(v_{11}\widetilde{\mathbf{x}}_c^1 + v_{12}\widetilde{\mathbf{x}}_c^2\right) \\ &= v_{11}^2\,\text{var}\left(\widetilde{\mathbf{x}}_c^1\right) + v_{12}^2\,\text{var}\left(\widetilde{\mathbf{x}}_c^2\right) + 2v_{11}v_{12}\,\text{c}(\widetilde{\mathbf{x}}_c^1, \widetilde{\mathbf{x}}_c^2)\end{aligned}$$



- Using the standardized version (scaled)

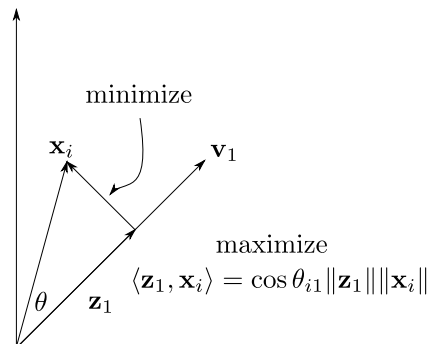
$$\text{var}(\mathbf{z}_1) = v_{11}^2 + v_{12}^2 + 2v_{11}v_{12} \times \text{r}(\widetilde{\mathbf{x}}_c^1, \widetilde{\mathbf{x}}_c^2)$$

New coordinates in the matricial framework (2)

- To find the new coordinates: find $\mathbf{v}_1$ such that $\text{var}(\mathbf{z}_1)$ is maximal

$$\text{var}(\mathbf{z}_1) \quad = \quad v_{11}^2 + v_{12}^2 + 2v_{11}v_{12} \times \text{r}(\widetilde{\mathbf{x}}_c^1, \widetilde{\mathbf{x}}_c^2)$$

- Constraint for a normed basis: $\|\mathbf{v}_1\|_2^2 = 1$
- This ensures that the new basis is of unitary scale, so that the information carried by the new axes can be compared



PCA as an optimization problem

- To find the first axis, find coefficients $\mathbf{v}_1$, s.t.

$$\begin{aligned}
\max_{\mathbf{v}_1, \|\mathbf{v}_1\|_2^2=1} \left\{ \text{var}(\mathbf{z}_1) \right\} &= \max_{\mathbf{v}_1, \|\mathbf{v}_1\|_2^2=1} \left\{ \text{var}(\mathbf{X}_c \mathbf{v}_1) \right\} \\
&= \max_{\mathbf{v}_1, \|\mathbf{v}_1\|_2^2=1} \left\{ \mathbf{v}_1 \left(\mathbf{X}_c' \mathbf{X}_c \right) \mathbf{v}_1' \right\} \\
&= \max_{\mathbf{v}_1, \|\mathbf{v}_1\|_2^2=1} \left\{ \mathbf{v}_1 \mathbf{S} \mathbf{v}_1' \right\}
\end{aligned}$$

- The solution of this optimization problem is explicit

$$\begin{aligned}
\mathbf{v}_1' \mathbf{v}_1 &= 1 \\
\mathbf{S} \mathbf{v}_1 &= \lambda_1 \mathbf{v}_1
\end{aligned}$$

- $\mathbf{v}_1$ (resp λ_1) is the first eigenvector (resp eigenvalue) of the **covariance** matrix

normed PCA as an optimization problem

- To find the first axis, find coefficients $\widetilde{\mathbf{v}}_1$, s.t.

$$\begin{aligned}
\max_{\widetilde{\mathbf{v}}_1, \|\widetilde{\mathbf{v}}_1\|_2^2=1} \left\{ \text{var}(\mathbf{z}_1) \right\} &= \max_{\widetilde{\mathbf{v}}_1, \|\widetilde{\mathbf{v}}_1\|_2^2=1} \left\{ \text{var}(\widetilde{\mathbf{X}}_c \widetilde{\mathbf{v}}_1) \right\} \\
&= \max_{\widetilde{\mathbf{v}}_1, \|\widetilde{\mathbf{v}}_1\|_2^2=1} \left\{ \widetilde{\mathbf{v}}_1 \left(\widetilde{\mathbf{X}}_c' \widetilde{\mathbf{X}}_c \right) \widetilde{\mathbf{v}}_1' \right\} \\
&= \max_{\widetilde{\mathbf{v}}_1, \|\widetilde{\mathbf{v}}_1\|_2^2=1} \left\{ \widetilde{\mathbf{v}}_1 \mathbf{R} \widetilde{\mathbf{v}}_1' \right\}
\end{aligned}$$

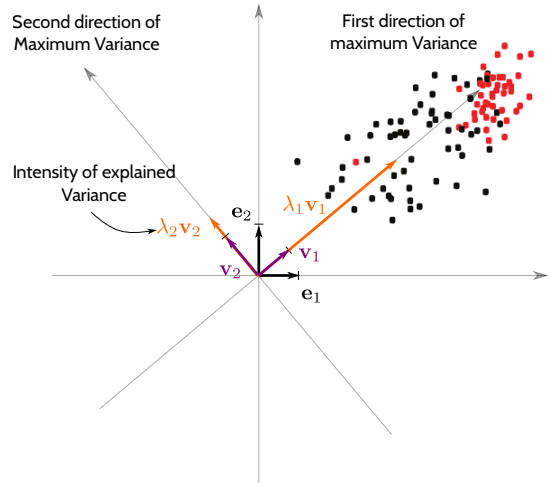
- The solution of this optimization problem is explicit

$$\begin{aligned}
\widetilde{\mathbf{v}}_1' \widetilde{\mathbf{v}}_1 &= 1 \\
\mathbf{R} \widetilde{\mathbf{v}}_1 &= \lambda_1 \widetilde{\mathbf{v}}_1
\end{aligned}$$

- $\widetilde{\mathbf{v}}_1$ (resp λ_1) is the first eigenvector (resp eigenvalue) of the **correlation** matrix

Eigen Representation of the data

- **S** contains the directions of maximal variance of the data
- $\mathbf{v}_1 \perp \mathbf{v}_2$ and are normed (unit variance)
- (λ_1, λ_2) quantify the amount of variance in each direction
- The eigen decomposition provides the best representation of the data in terms of variance
- Its the linear transform that makes the new set of coordinates diagonal



Quality of the representation

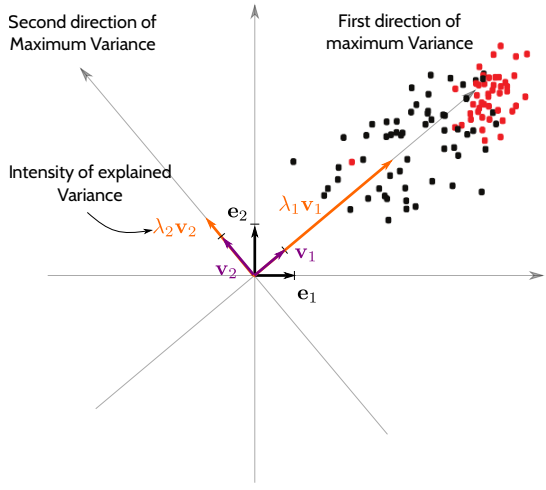
- Eigenvalues quantify the inertia of the dataset:

$$I_T(X) = \sum_{k=1} I_k(X) = \sum_{k=1}^{K} \lambda_k$$

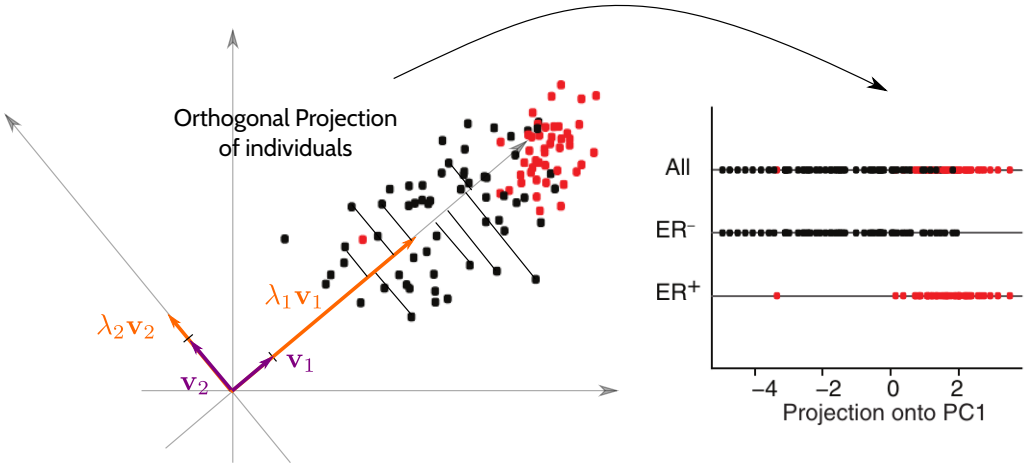
- Percent of explained variance:

$$\text{Contrib}_k = \frac{\lambda_k}{\sum_{\ell=1}^{K} \lambda_\ell}$$

$$\text{Contrib}_{1:k} = \frac{\sum_{h=1}^{k} \lambda_h}{\sum_{\ell=1}^{K} \lambda_\ell}$$



Representation of individuals in the new coordinates



The new coordinates for individuals are $\left(\mathbf{x}_i - \overline{\mathbf{x}}\right)\mathbf{v}_k$

Outline

Outline

- In a first step we changed coordinates for better representation
- From 2D to 2D, there is no dimension reduction !
- The approach is generalized from p variables to K principal components

$$\mathbf{z}_k = \sum_{j=1}^{p} v_{kj} \widetilde{\mathbf{x}}_c^j = \mathbf{X}_c \mathbf{v}_1$$

- Intuition: if v_{kj} is high, variable j highly contributes to principal component $\mathbf{z}_k$
- From p to $K(=2)$ the information was compressed

General Case with K principal components

- $\mathbf{V}_{[p\times K]} = \left[\mathbf{v}_1, \ldots, \mathbf{v}_K\right]$, the eigen vectors of the covariance matrix

$$\mathbf{S}_{p\times p} = \frac{1}{n}\mathbf{X}'\mathbf{X} = \frac{1}{n}\sum_{k=1}^{K}\lambda_k \mathbf{v}_k \mathbf{v}_k'$$

- $\mathbf{U}_{[n\times K]} = \left[\mathbf{u}_1, \ldots, \mathbf{u}_K\right]$, the eigen vectors of the Gram matrix

$$\mathbf{G}_{n\times n} = \frac{1}{p}\mathbf{X}\mathbf{X}' = \frac{1}{p}\sum_{k=1}^{K}\lambda_k \mathbf{u}_k \mathbf{u}_k'$$

- Then we have

$$\begin{aligned}(\mathbf{X}\mathbf{X}')\mathbf{u}_k &= \sqrt{\lambda_k}\mathbf{X}\mathbf{v}_k = \lambda_k \mathbf{u}_k \\ (\mathbf{X}'\mathbf{X})\mathbf{v}_k &= \sqrt{\lambda_k}\mathbf{X}'\mathbf{u}_k = \lambda_k \mathbf{v}_k\end{aligned}$$

Low-rank approximation of X

- The rank of a matrix (r^*) is the number of linearly independent columns (unknown in practice)
- From a statistical perspective, it is the number of independent coordinates that can describe a dataset
- The initial dataset can be rewritten such that

$$\mathbf{X} = \mathbf{U}_{n \times r^*} \mathbf{V}'_{r^* \times p} = \sum_{k=1}^{r^*} \sqrt{\lambda_k} \mathbf{u}_k \mathbf{v}'_k$$

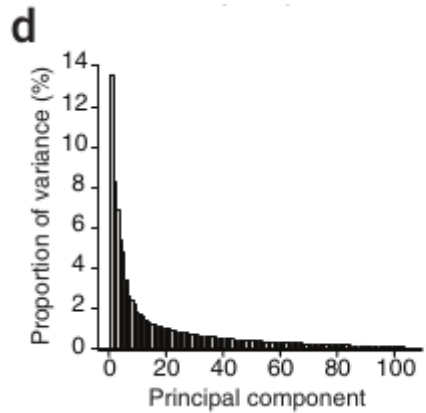
- Since the rank is unknown, we select a number of components K, and then:

$$\mathbf{X} \simeq \mathbf{U}_{n \times K} \mathbf{V}'_{K \times p} = \sum_{k=1}^{K} \sqrt{\lambda_k} \mathbf{u}_k \mathbf{v}'_k$$

- It is called the low-rank approximation of $\mathbf{X}$

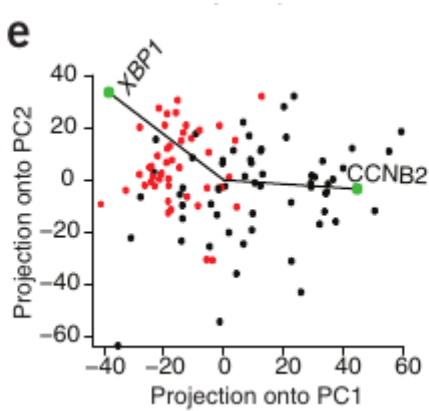
PCA on the complete ER dataset - 1

- First examples on 2 genes without dimension reduction
- PCA on the $p = 8534$ genes, $n = 105$ individuals
- $K_{\max} = 8534$ possible eigenvectors
- $\text{Contrib}_{1:2} \simeq 22\%$
- $\text{Contrib}_{1:63} \simeq 90\%$
- $\text{Contrib}_{1:104} \simeq 100\%$
- Choosing 104 eigenvectors reduces the dimension without too much loss
- Dimension reduction : from 8534 original variables to 104 new variables



PCA on the complete ER dataset - 2

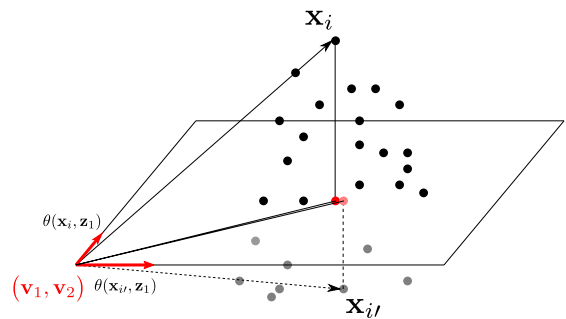
- Represent the data in the new coordinates (PCs)
- In the example the clusters (ER+/ER-) are more separable in the new representation
- Identify the contribution of genes to the axes
- Essential to interpret the new representation



Quality of the representation of individuals

- An individual $\mathbf{x}_i$ is well represented if it is close to the axis $\mathbf{z}_k$
- Geometrically, $\mathbf{x}_i - \overline{\mathbf{x}}$ is colinear to $\mathbf{z}_k$
- Compute

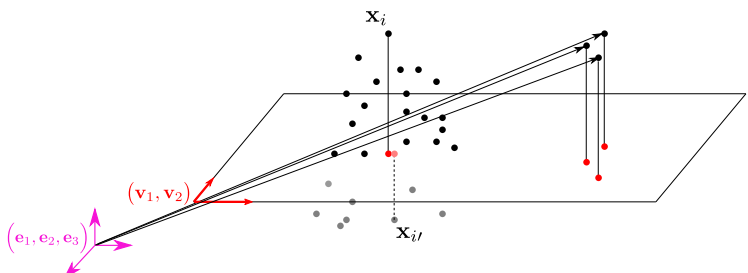
$$\cos^2 \theta(\mathbf{x}_i - \overline{\mathbf{x}}, \mathbf{z}_k) = \frac{\Big(\big(\mathbf{x}_i - \overline{\mathbf{x}}\big)\mathbf{v}_k\Big)^2}{\|\mathbf{x}_i - \overline{\mathbf{x}}\|^2 \|\mathbf{v}_k\|^2}$$



Contribution of individuals to the representation

The contribution of a $\mathbf{x}_i$ is the proportion of carried by $\mathbf{x}_i$

$$\text{contr}(\mathbf{x}_i, \mathbf{z}_k) = \frac{\Big(\big(\mathbf{x}_i - \overline{\mathbf{x}}\big)\mathbf{v}_k\Big)^2}{n\lambda_k}$$



Properties of Principal components: the variable point of view

- Start with p correlated (redundant) variables $\widetilde{\mathbf{X}}_c = \left[\tilde{\mathbf{x}}_c^1, \ldots, \tilde{\mathbf{x}}_c^p \right]$ with

$$\mathbf{R}_{p\times p} = \begin{bmatrix} \mathrm{r}(\mathbf{x}^1, \mathbf{x}^1) & \ldots & \mathrm{r}(\mathbf{x}^{j'}, \mathbf{x}^j) \\ & \ddots & \\ \mathrm{r}(\mathbf{x}^j, \mathbf{x}^{j'}) & \ldots & \mathrm{r}(\mathbf{x}^p, \mathbf{x}^p) \end{bmatrix} = \frac{1}{n}\widetilde{\mathbf{X}}_c'\widetilde{\mathbf{X}}_c = \sum_{k=1}^{K} \lambda_k \mathbf{v}_k \mathbf{v}_k'$$

- Get K new uncorrelated (non redundant) variables $\mathbf{Z} = \left[\mathbf{z}^1, \ldots, \mathbf{z}^K \right]$

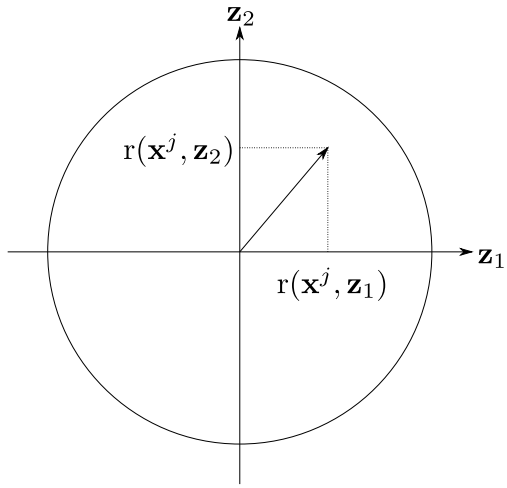
Correlation Circle

- Components are independent of variance with $\text{var}(\mathbf{z}_k) = \lambda_k$

$$\mathbf{S}_Z = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_K \end{bmatrix}$$

- Contribution of variables to axis:

$$\begin{aligned} \text{c}(\mathbf{x}^j, \mathbf{z}_k) &= (\mathbf{x}^j)' \mathbf{u}_k = \lambda_k v_{jk} \\ &= \text{r}(\mathbf{x}^j, \mathbf{z}_k) \text{ for normed PCA} \\ \text{c}(\mathbf{X}, \mathbf{Z}) &= \mathbf{S}_Z \mathbf{V} \end{aligned}$$



Quality of representation of variables in PCs

- Check the quality of representation of variable $\mathbf{x}^j$ on PC k

$$I_T(\mathbf{X}) = \sum_{j=1}^{p} \sum_{s=1}^{r} \mathrm{r}^2(\mathbf{x}^j, \mathbf{z}_s) \quad \text{for normed PCA}$$

- Correlation circle:

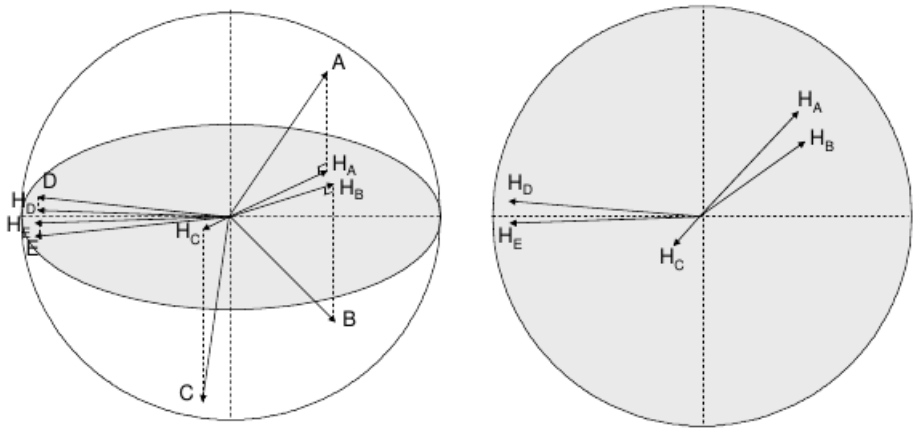
$$\cos^2\left(\theta\{\mathbf{x}^j, \mathbf{z}_k\}\right) = \frac{\mathrm{r}^2(\mathbf{x}^j, \mathbf{z}_k)}{\sum_{s=1}^{r} \mathrm{r}^2(\mathbf{x}^j, \mathbf{z}_s)}$$

- Only variables with high $\cos^2$ can be interpreted !
- Contribution of variable $\mathbf{x}^j$

$$\mathrm{contr}(\mathbf{x}^j, \mathbf{z}_k) = \frac{\mathrm{r}^2(\mathbf{x}^j, \mathbf{z}_k)}{\lambda_k}$$

Quality of representation of variables in PCs

Check the quality of representation of variables, close variables are not necessarily similar



Outline

Summary

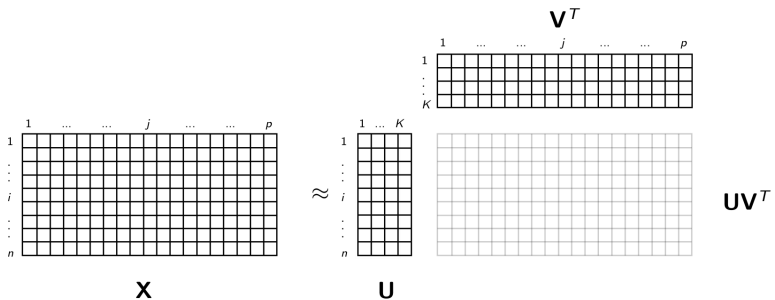
- PCA is the most widely used linear dimension reduction method
- It is based on a change in coordinates to represent the data in a way that preserves the variability of the data
- The new coordinates are provided by the eigenvectors of the empirical variance matrix
- Check the percentage of explained variance and choose the number of components accordingly
- Check the quality of representation of variables to interpret the axes
- Interpret the projection of individuals at the end
- Why does PCA make cluster more visible ?

Matrix factorization: $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$

Cells: $\mathbf{U} \in \mathbb{R}^{n \times K}$

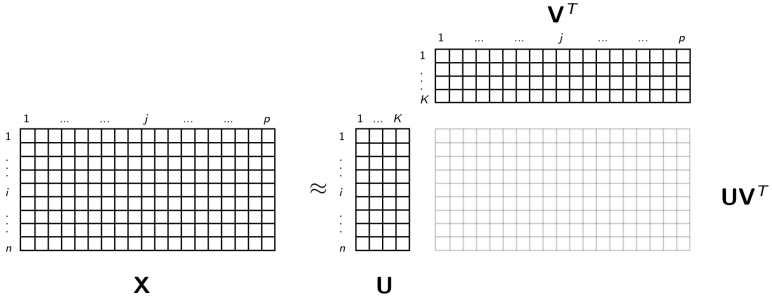
Genes: $\mathbf{V} \in \mathbb{R}^{p \times K}$

} Low dimensional representation



$\rightarrow$ Low-rank representation of $\mathbf{X}$

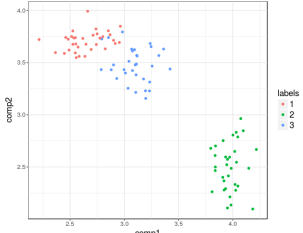
Matrix factorization: $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$



Data visualization: U

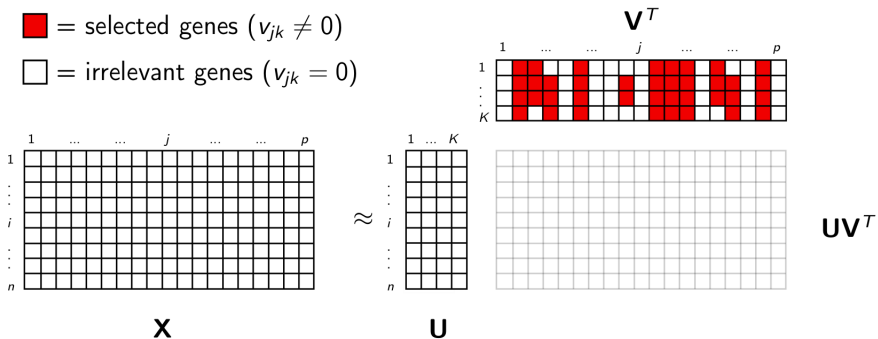
scatter plot $(u_{i1}, u_{i2})_{i=1:n}$

Embeddings



How to interpret the axes ?

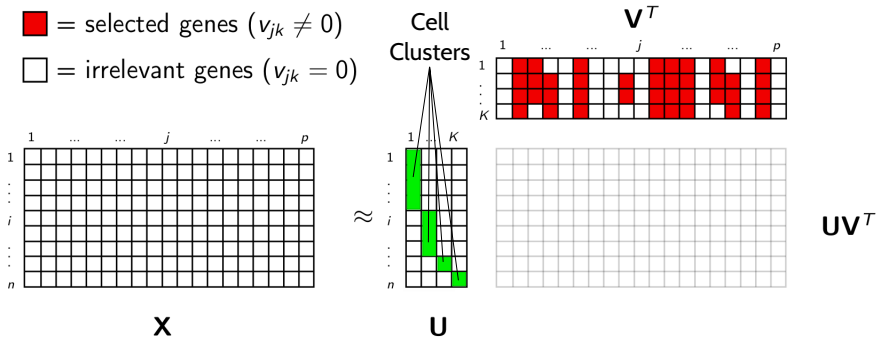
- When genes contributes poorly to axis, their contribution can be put to zero



- Selected genes can be interpreted in terms of signature.

How to cluster cells in terms of selected variables

- When signatures are selected in **V**, this can be used to create clusters of cells in **U**



- Compression allows to exhibit variables that make clusters more detectable

Towards embedding methods

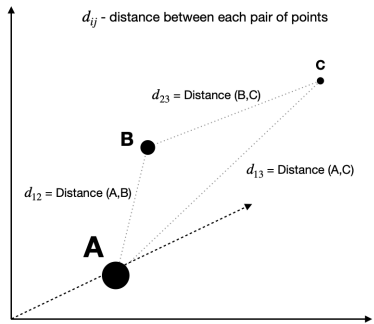
- PCA is based on the duality between the between-variables distance $\mathbf{S} = \mathbf{X}'\mathbf{X}/n$ and the between individuals distance $\mathbf{G} = \mathbf{X}\mathbf{X}'/p$
- $\mathbf{U}$ provides the new coordinates for the individuals
- $\mathbf{V}$ provides the new coordinates for the variables
- Creating a new representation thanks to a linear transform $\mathbf{Z} = \mathbf{X}\mathbf{V}'$ ensures the same transform for each point
- The linear nature of the transform ensures interpretability of PCA
- In the end, data vizualization focuses on the representations of individuals, called embeddings.
- Considering embedding allows to extend the notion of dimension reduction to other frameworks

A primer with Multidimensional Scaling

- In many situations only the distance $d_{ii'}$ between individuals (i, i') is available
- The objective of MDS is to find new coordinates $\mathbf{u}_1, \ldots, \mathbf{u}_n$ that minimize:

$$\sum_{i,i'} \left(d_{ii'} - \|\mathbf{u}_i - \mathbf{u}_{i'}\|^2 \right)^2$$

- The information regarding the variables is not considered (not available)

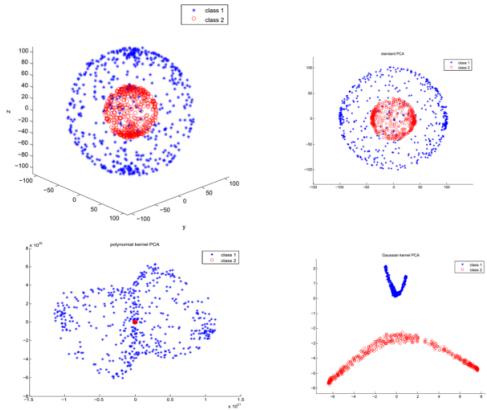


Extending the notion of distance with kernels

- Linear methods are mainly based on euclidean distances
- These distances depend on a dot product
- This dot product can be generalized by the so-called kernel

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_{i'}) \rangle$$

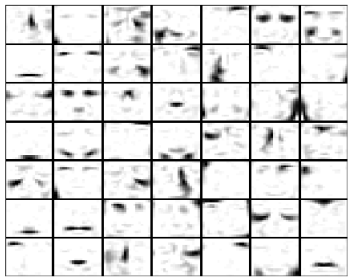
- ϕ is called the feature map and is unknown
- Grounds most non linear methods (kernel-PCA, kernel MDS, etc)

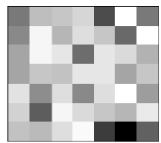


Accounting for particular characteristics of data

When data are counts, introduce a non-negativity constraint and use NMF

$$\underbrace{X(:,j)}_{j\text{th facial image}} \approx \sum_{k=1}^{r} \underbrace{W(:,k)}_{\text{facial features}} \underbrace{H(k,j)}_{\substack{\text{importance of features} \\ \text{in } j\text{th image}}} = \underbrace{WH(:,j)}_{\substack{\text{approximation} \\ \text{of } j\text{th image}}} .$$

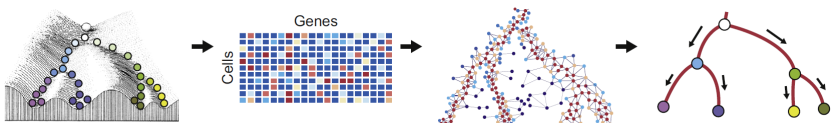




Outline

Beyond Linear projections

- Linear methods are powerful for planar structures
- High dimensional datasets are characterized by multiscale properties (local / global structures)
- May not be the most powerful for manifolds
- Non Linear projection methods aim at preserving local characteristics of distances



Stochastic Neighbor Embedding [?]

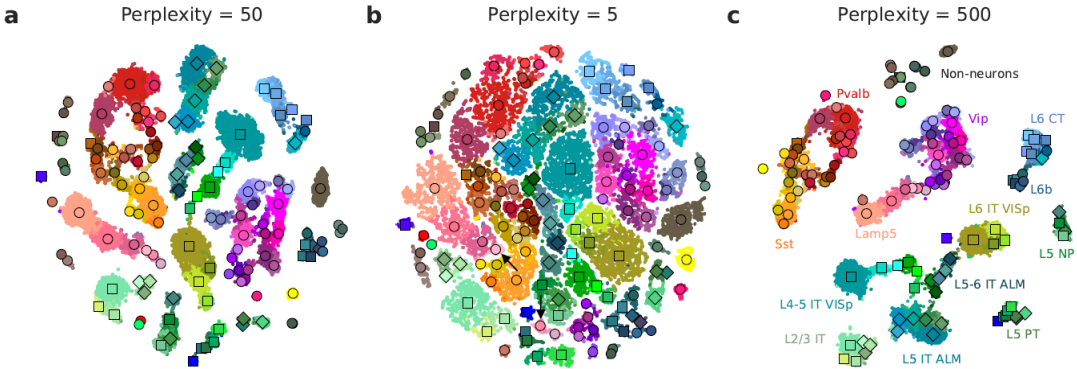
- $(x_1, \ldots, x_n)$ are the points in the high dimensional space $\mathbb{R}^p$,
- Consider a similarity between points:

$$p_{i|j} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_k - x_j\|^2/2\sigma_k^2)}, \; p_{ij} = (p_{i|j} + p_{j|i})/2N$$

- σ smooths the data (linked to the regularity of the target manifold)
- σ is chosen such that the entropy of p is fixed to a given value of the so-called perplexity

$$\exp\left(-\sum_{ij} p_{ij} \log(p_{ij})\right)$$

Visual inspection of the influence of σ[?]



tSNE and Student / Cauchy kernels

- Consider $(y_1, \ldots, y_n)$ are points in the low dimensional space $\mathbb{R}^2$
- Consider a similarity between points in the new representation:

$$q_{i|j} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_k - y_j\|^2)}$$

- Robustify this kernel by using Student(1) kernels (ie Cauchy)

$$q_{i|j} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_i - y_k\|^2)^{-1}}$$

Optimizing tSNE

- Minimize the KL between p and q so that the data representation minimizes:

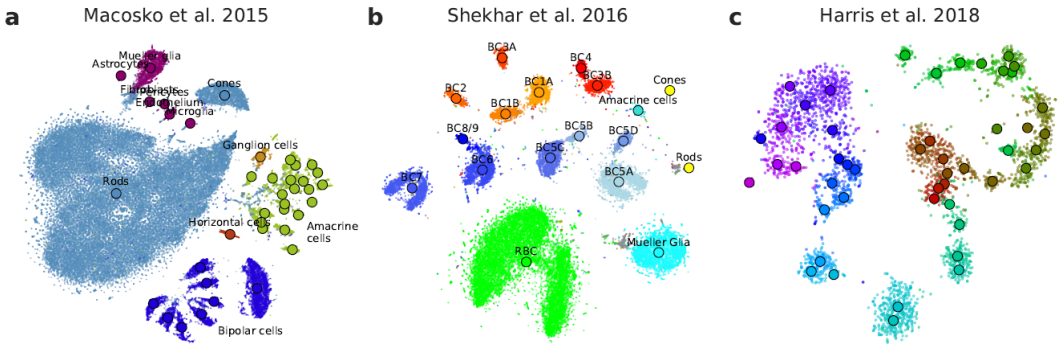
$$C(y) = \sum_{ij} KL(p_{ij}, q_{ij})$$

- The cost function is not convex

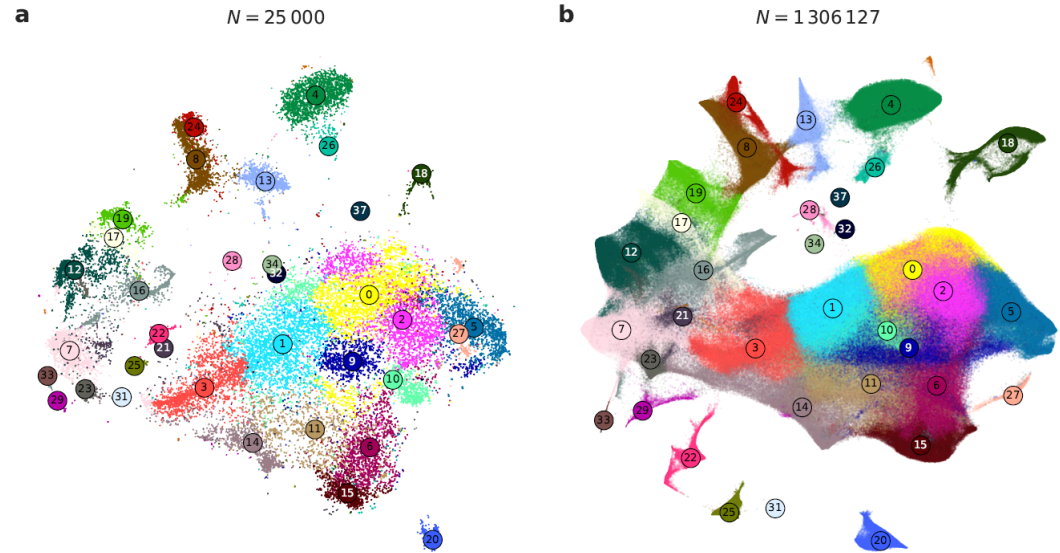
$$\left[\frac{\partial C(y)}{\partial y}\right]_i = \sum_j (p_{ij} - q_{ij})(y_i - y_j)$$

- Interpreted as the resultant force created by a set of springs between the map point y_i and all other map points $(y_j)_j$. All springs exert a force along the direction $(y_i - y_j)$.
- $(p_{ij} - q_{ij})$ is viewed as a stiffness of the force exerted by the spring between y_i and y_j.

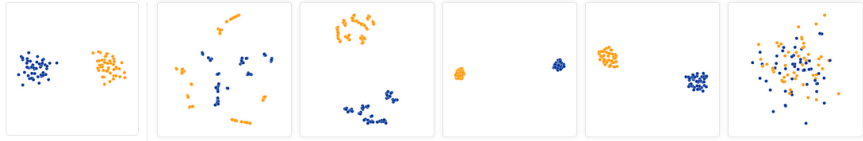
tSNE examples on single cell RNASeq data 1 [?]



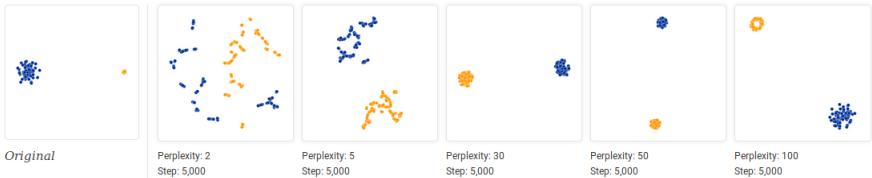
tSNE examples on single cell RNASeq data 1 [?]



Effect of Hyperparameters : Perplexity

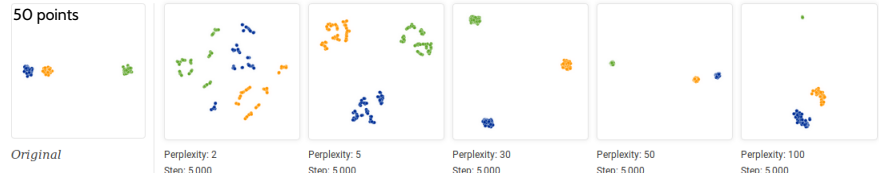


tSNE does not account for heteroscedasticity

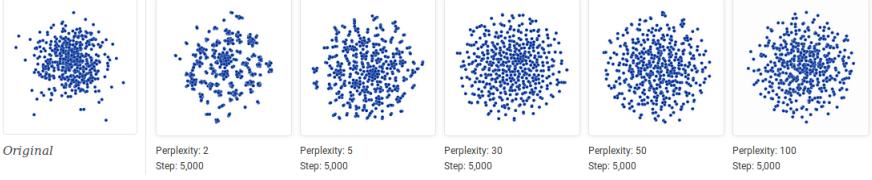


https://distill.pub/2016/misread-tsne/

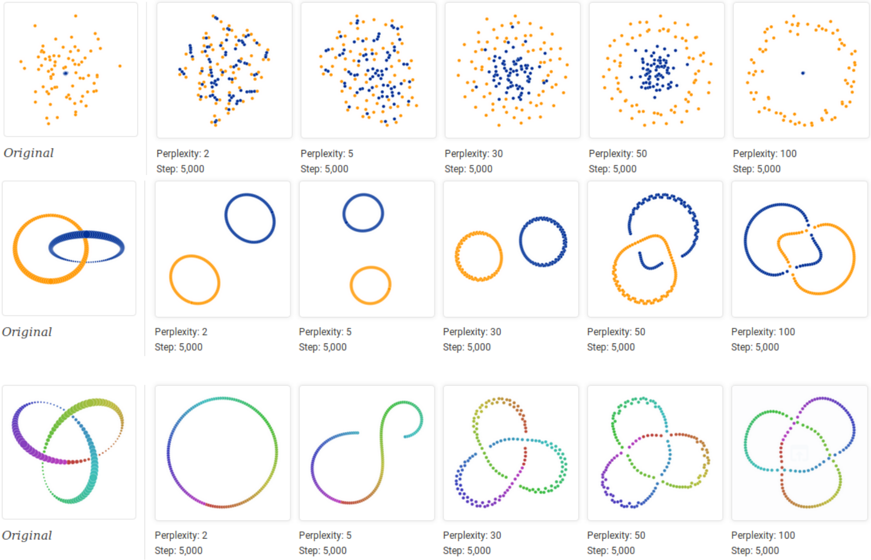
tSNE does not account for between-cluster distance



What about random noise ?



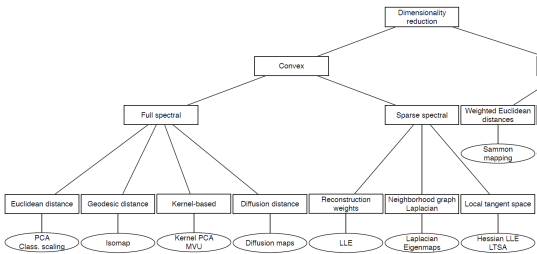
Catching Complex Geometries



Properties of t-SNE

- good at preserving local distances (intra-cluster variance)
- not so good for global representation (inter-cluster variance)
- hence good at creating clusters of points that are close, but bad at positionning clusters wrt each other
- preprocessing very important : initialize with PCA and feature selection plus log transform (non linear transform)
- percent of explained variance ? interpretation of the q distribution ?

A taxonomy of Dimension Reduction Methods [?]



Technique	*Parametric*	*Parameters*	*Computational*	*Memory*
PCA	yes	none	$O(D^3)$	$O(D^2)$
Class. scaling	no	none	$O(n^3)$	$O(n^2)$
Isomap	no	k	$O(n^3)$	$O(n^2)$
Kernel PCA	no	$\kappa(\cdot,\cdot)$	$O(n^3)$	$O(n^2)$
MVU	no	k	$O((nk)^3)$	$O((nk)^3)$
Diffusion maps	no	σ, t	$O(n^3)$	$O(n^2)$
LLE	no	k	$O(pn^2)$	$O(pn^2)$
Laplacian Eigenmaps	no	k, σ	$O(pn^2)$	$O(pn^2)$
Hessian LLE	no	k	$O(pn^2)$	$O(pn^2)$
LTSA	no	k	$O(pn^2)$	$O(pn^2)$
Sammon mapping	no	none	$O(in^2)$	$O(n^2)$
Autoencoders	yes	net size	$O(inw)$	$O(w)$
LLC	yes	m, k	$O(imd^3)$	$O(md)$
Manifold charting	yes	m	$O(imd^3)$	$O(md)$

Conclusions of a comparative study [?]

- local methods suffer from the choice of the smoothing (neighborhood) parameter
- Kernel PCA suffers from the choice of the Kernel to correctly approximate the manifold.
- Setting the optimization problem is the key (convex or not), trivial solutions, local optima, computationally feasible
- nonlinear techniques for dimensionality reduction are, despite their large variance, often not capable of outperforming traditional linear techniques such as PCA.

Useful links

- https://towardsdatascience.com/
- PCA for datascience
- Link to a tuto on dot products
- Wiki for Linear Transforms
- Book for the introduction to machine learning (C.-A. Azencott)
- Book for the introduction to machine learning (James, Witten, Hastie, Tibshirani)
- PCA in ecology http://pbil.univ-lyon1.fr/ade4/
- PCA in general http://factominer.free.fr/index_fr.html

References

[1] LJP Van der Maaten, EO Postma, and HJ Van den Herik. Dimensionality reduction: A comparative review. *TiCC*, 2009.

[2] J. G. Greener, S. M. Kandathil, L. Moffat, and D. T. Jones. A guide to machine learning for biologists. *Nat Rev Mol Cell Biol*, 23(1):40–55, 01 2022.

[3] Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *bioRxiv*, 2018.

[4] M. Ringnér. What is principal component analysis? *Nat Biotechnol*, 26(3):303–304, Mar 2008.

[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Outline

Expectation / Variance for matrices

- Given $Y_i \in \mathbb{R}^p$, $A \in \mathbb{R}^{q \times p}$,
$$\mathbb{E}(AY_i) = A \times \mathbb{E}(Y_i),$$
- The variance of a linear combination of Y
$$\mathbb{V}(AY_i) = A\mathbb{V}_{p\times p}(Y_i)A',$$

Outline

Decomposition of $\mathbb{R}^p$ into orthogonal subspaces

- Let us consider p orthogonal subspaces $\left(E_k\right)_{k=1,p}$ each subspace spanned by an individual axis (dim 1):

$$\mathbb{R}^p = \bigoplus_{k=1}^{p} E_k,$$

- Orthogonal projection of $X_i \in \mathbb{R}^p$ on a subspace $E_k = \text{vect}(Z_k)$

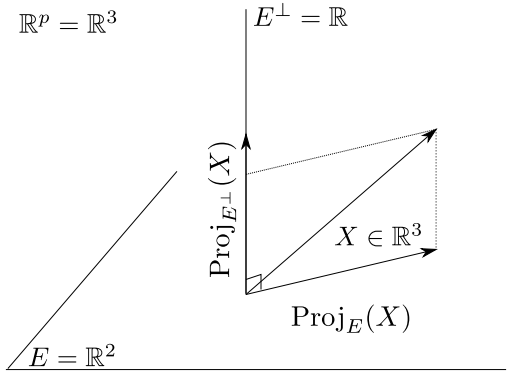
$$\text{Proj}_{E_k}(X_i) = X_i V_k \in \mathbb{R}$$

- The inertia of X wrt E_k measures the proximity of E_k from X

$$I_{E_k}(X) = \frac{1}{n} \sum_{i=1}^{n} \|X_i - \text{Proj}_{E_k}(X_i)\|_2^2$$

- Let $E_k^{\perp}$ denotes the orthogonal complement of subspace E_k.

Pythagore - Huyguens Theorem



$$I_T(X) = I_E(X) + I_{E^\perp}(X) = I\Big(\mathrm{Proj}_E(X)\Big) + I\Big(\mathrm{Proj}_{k^\perp}(X)\Big)$$

Construction of principal components (PC)

- Resume the data X by a new dataset $Z_{n\times K}$, $K \leq p$ and K fixed
- The new axis spans the 1-dim subspaces $\Big(E_k = \text{vect}(Z_k)\Big)_k$

$$\forall k, k', \quad E_k \perp E_{k'}$$

- $Z = [Z_1, \ldots, Z_K]$ constitute independent PCs (easy interpretation)
- $Z_k \in \mathbb{R}^n$ is defined as a linear combination of the variables

$$Z_k = XV_k, \quad V_k = \big(V_{jk}\big)_j \in \mathbb{R}^p$$

- $V_{p\times K} = [V_1, \ldots, V_K]$ is the matrix of contributions (weights) of variables $\big(X^j\big)_j$

$$Z_{n\times K} \quad = \quad X_{n\times p} V_{p\times K}$$

Decomposition of the Inertia on the PCs

$$
\begin{aligned}
I_T(X) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\|X_i - \mathrm{Proj}_{E_k}(X_i) + \mathrm{Proj}_{E_k}(X_i)\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\|X_i - \mathrm{Proj}_{E_k}(X_i)\|^2 + \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\|\,\mathrm{Proj}_{E_k}(X_i)\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\|X_i - Z_{ik}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\|Z_{ik}\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\|X_i - X_i V_k\|^2 + \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\|X_i V_k\|^2
\end{aligned}
$$

Orthogonal Components with maximal variance

- We want to resume the variability of the dataset
- Find the PCs that explain the maximum of the observed variance:

$$\frac{1}{n}\sum_{i=1}^{n} \| \text{Proj}_{E_k}(X_i)\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|Z_{ik}\|^2 = \frac{1}{n} V_k' \Big(X'X \Big) V_k = \frac{1}{n} V_k' \Sigma V_k$$

- The optimization scheme is iterative, and for the kth PC:

$$\widehat{V}_k = \underset{V \in \mathbb{R}^p, \|V\|_2^2 = 1}{\arg\max} \left(\frac{1}{n} V'X'XV \right) \quad \text{with } Z_k \perp (Z_1, \ldots, Z_{k-1})$$

Constrained optimization

- To account for the orthogonality constraint, we introduce the Lagrange multipliers

$$\begin{aligned}
\mathcal{L}(V,\mu) &= \frac{1}{n} V'X'XV - \mu\Big(V'V - 1\Big) \\
\frac{\partial L}{\partial \mu} &= V'V - 1 \\
\frac{\partial L}{\partial V} &= 2X'XV - \mu V
\end{aligned}$$

- Which gives the following solution

$$\begin{aligned}
V'V &= 1 \\
X'XV &= \mu V
\end{aligned}$$

- The optimal solution is provided by the eigenvectors of the covariance matrix Σ

Spectral decomposition of symmetric real matrices

- Let $A \in \mathbb{R}^{n,n}$ a symmetric real matrix
- Spectral decomposition theorem: there exists $\lambda_1 \geq \ldots \geq \lambda_n \in \mathbb{R}$ and an orthogonal basis $\{U_1, \ldots, U_n\}$ of $\mathbb{R}^n$ such that

$$A = \sum_{k=1}^{n} \lambda_k U_k U_k'$$

- The spectral decomposition can also be written:

$$A = U \operatorname{diag}(\lambda_1, \ldots, \lambda_n) U'$$

Positive Semi-Definite Matrices

- A symmetric real matrix is positive semi-definite (sdp) if

$$\forall \in \mathbb{R}^n,\ x'Ax \geq 0$$

- Semi-Definite positiveness is equivalent to $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$, since

$$x'Ax = \sum_{k=1}^{n} \lambda_k \langle x, U_k \rangle^2$$

- For any $n \times p$ matrix A, the matrices $A'A$ and AA' are symmetric positive semidefinite

Singular Value Decomposition Theorem

- Any matrix $A \in \mathbb{R}^{n,p}$ of rank r can be decomposed as

$$A = \sum_{k=1}^{r} \mu_k U_k V_k'$$

- $r = \text{rank}(A)$
- $\mu_1 \geq \ldots \geq \mu_r > 0$
- $\{\mu_1^2, \ldots, \mu_r^2\}$ are the non-zero eigenvalues of $A'A$ and of AA'
- $\{\mu_1, \ldots, \mu_r\}$ are called the singular values of A
- $\{U_1, \ldots, U_r\}$ and $\{V_1, \ldots, V_r\}$ are two orthonormal families of $\mathbb{R}^n$ and $\mathbb{R}^p$ such that:

$$AA'U_k = \mu_k^2 U_k, \quad A'AV_k = \mu_k^2 V_k$$

Singular Value Decomposition of $X'X$ and XX'

- $(U_1, \ldots, U_K)$, the eigen vectors of the Gram matrix

$$G_{n\times n} = \frac{1}{p} XX' = \frac{1}{p} \sum_{k=1}^{K} \mu_k^2 U_k U_k'$$

- $(V_1, \ldots, V_K)$, the eigen vectors of the covariance matrix

$$\Sigma_{p\times p} = \frac{1}{n} X'X = \frac{1}{n} \sum_{k=1}^{K} \mu_k^2 V_k V_k'$$

- Then we have

$$\begin{aligned} (XX')U_k &= \mu_k X V_k = \mu_k^2 U_k \\ (X'X)V_k &= \mu_k X' U_k = \mu_k^2 V_k \end{aligned}$$

Low-rank approximation of X (1)

- $X \in \mathbb{R}^{n,p}$, s.t. $\text{rank}(X) = r$, there exists
  $\rightarrow \mu_1 \geq \ldots \geq \mu_r > 0$, with $D = \text{diag}(\mu_1, \ldots, \mu_r)$,
  $\rightarrow \{\mu_1, \ldots, \mu_r\}$, are the singular values of X
  $\rightarrow$ two orthogonal matrices $\widetilde{U} \in \mathbb{R}^{n \times r}$ and $\widetilde{V} \in \mathbb{R}^{p \times r}$ with

$$\widetilde{U}'\widetilde{U} = I_r, \quad \widetilde{V}'\widetilde{V} = I_r,$$

$$U = \widetilde{U}D, \quad V = \widetilde{V}D,$$

- Such that

$$X = UV' = \widetilde{U}D\widetilde{V}' = \sum_{k=1}^{r} \mu_k \widetilde{U}_k \widetilde{V}_k'$$

- Then we have

$$X'\widetilde{U}_k = \mu_k \widetilde{V}_k, \quad X\widetilde{V}_k = \mu_k \widetilde{U}_k$$

Low Rank approximation of X (2)

- If $\text{rank}(X) = r$ (unknown), in practice we choose $K \leq p$ to provide a "low-rank" approximation of X.
- Denoting $\widehat{X}_K = U_{1:K} V'_{1:K}$ this approximation of $\text{rank}(\widehat{X}_K) = K$
- PCA can be restated as the approximation of X st

$$\|X - \widehat{X}_K\|_F^2 = \min_{B \in \mathcal{M}_{n,K}, rk(B)=K} \|X - B\|_F^2 = \sum_{k=K+1}^{r} \mu_k^2$$

- PCA provides the best low-rank approximation for the Frobenius norm

$$\widehat{X}_K = \underset{B \in \mathcal{M}_{n,K}, rk(B)=K}{\arg\min} \|X - B\|_F^2$$